



دانشکده فناوری های نوین پزشکی

راهنمای استفاده از خدمات مرکز پردازش سریع دانشکده

فناوری های نوین پزشکی اصفهان

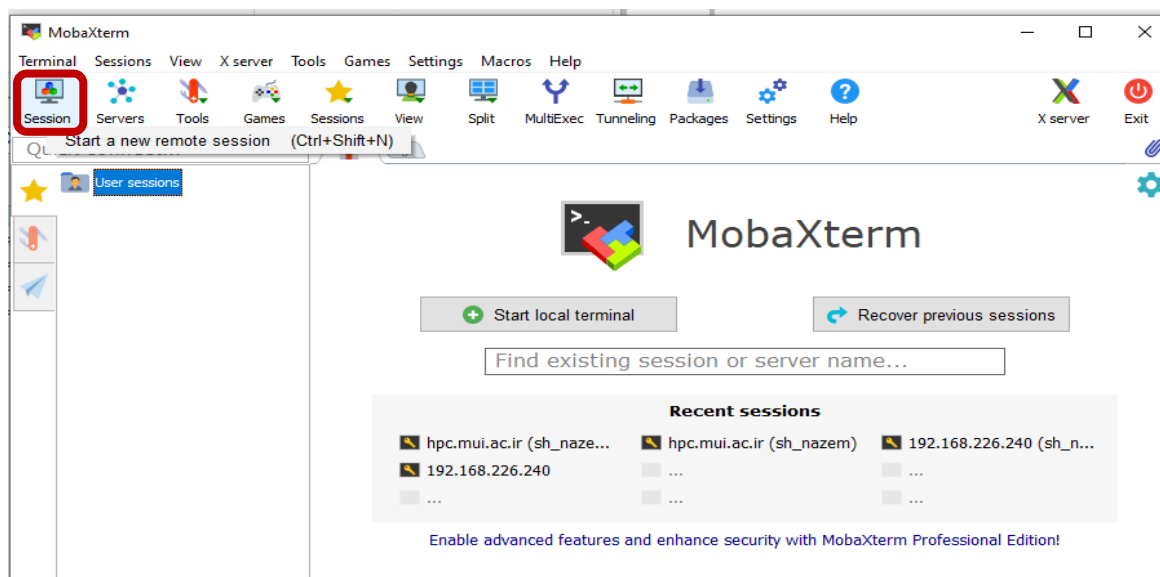
جهت دسترسی از راه دور به سامانه و اجرای دستورات و نرم افزارهای مورد نظر، این دستورات باید از طریق یک پروتکل ارتباطی به سامانه منتقل شوند. نرم افزارهای منبع باز متعددی برای این منظور از قبیل Putty و یا MobaXTerm در سیستم عامل ویندوز وجود دارند و در سیستم عامل لوکس از خود محیط ترمینال برای دسترسی استفاده می شود.

از نرم افزار MobaXTerm برای ارسال و دریافت فایل نیز بعنوان یک پروتکل انتقال فایل می توان استفاده کرد. این نرم افزار را می توانید از طریق لینک زیر دانلود کنید.

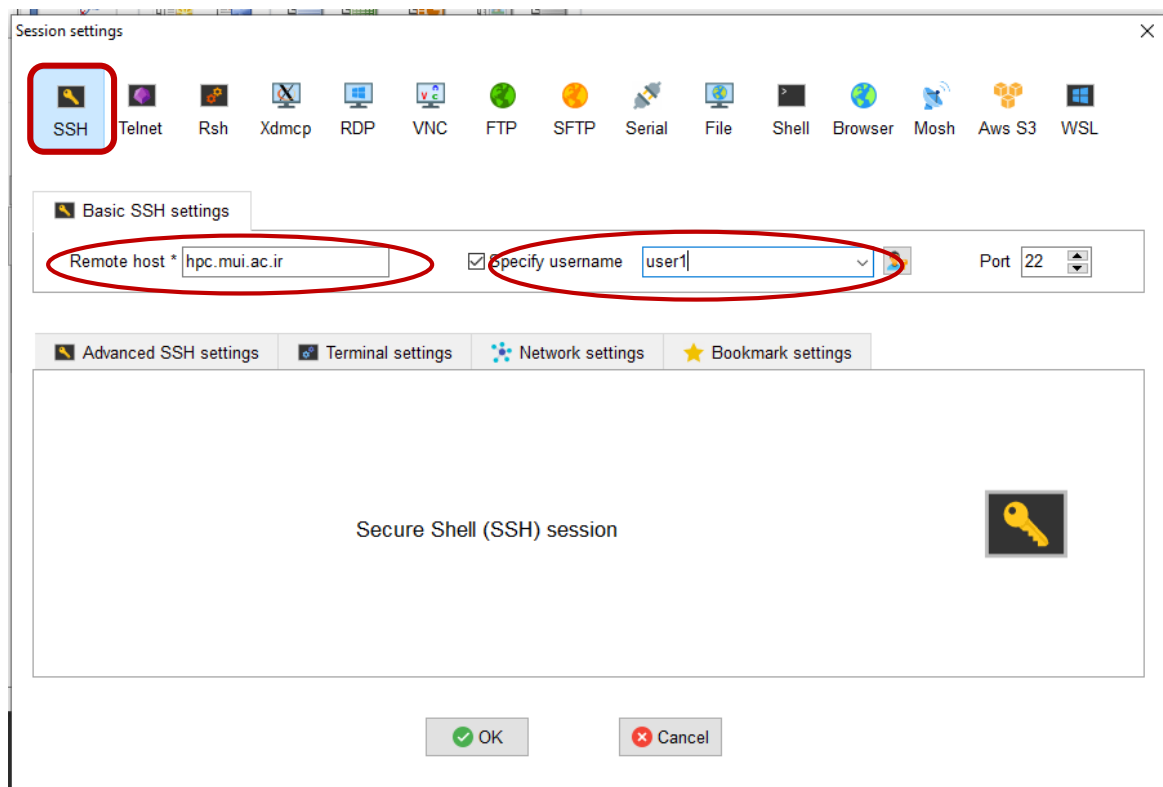
<https://mobaxterm.mobatek.net/>

پس از دانلود و نصب برنامه باید یک session جدید ایجاد نمایید. توجه داشته باشید که قبل از ایجاد session جدید اگر قصد دارید از داخل دانشگاه به ماشین مجازی دسترسی داشته باشید کافیست فقط به شبکه داخلی دانشگاه متصل باشید. کاربرانی که قصد دسترسی به ماشین مجازی از خارج دانشگاه را دارند، بعد از اتصال به اینترنت باید از طریق vpn خود به شبکه داخلی دانشگاه متصل شوند.

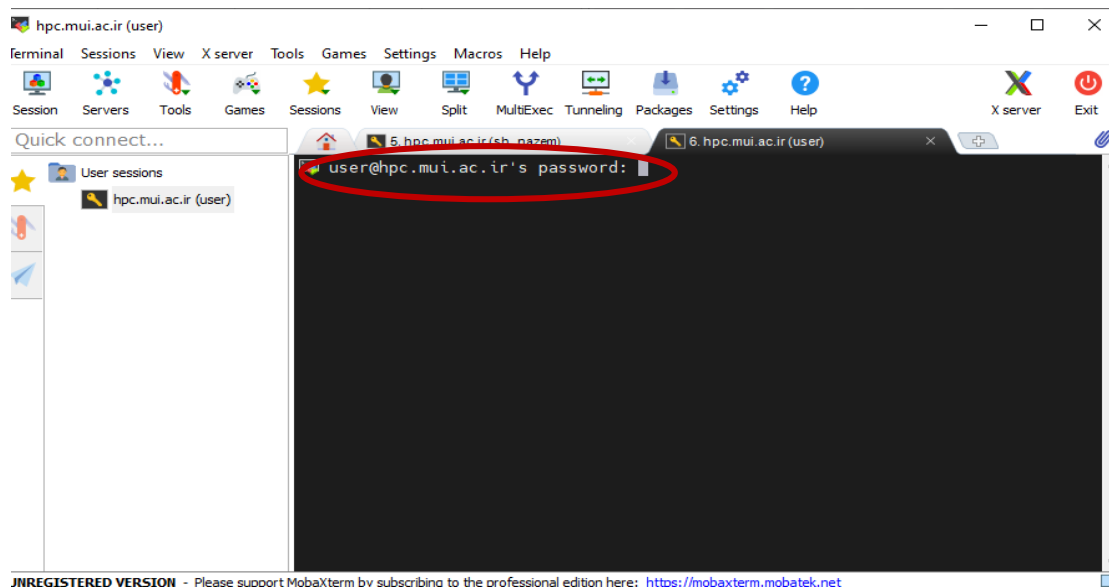
برای ایجاد session جدید مطابق با شکل زیر در داخل برنامه گزینه session را انتخاب نمایید.



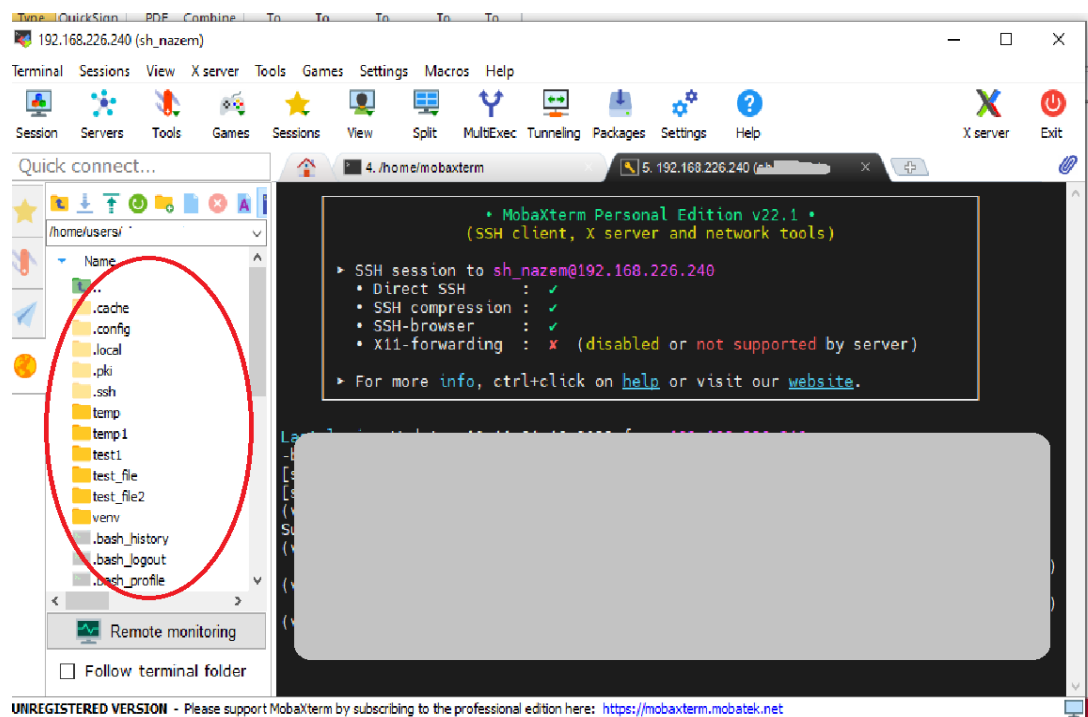
در داخل session جدید گزینه SSH را انتخاب کرده و در آن باید مشخصات سیستم که میخواهید به آن وصل شوید و نام کاربری که از سامانه دریافت کرده اید را وارد کنید. بدین منظور در قسمت remote host آدرس سیستم زیست ایران hpc.mui.ac.ir (یا 192.168.226.240) را وارد کرده و در قسمت specify username نام کاربری دریافتی خود را وارد نمایید بر روی ok کلیک نمایید.



در مرحله بعد از شما رمز عبور خواسته می شود. به حروف بزرگ و کوچک رمز دریافتی دقت کنید و توجه داشته باشید که در هنگام ورود رمز عبور هیچ کارکتری نشان داده نمی شوند.



در سمت راست صفحه می توانید فایل های موجود بر روی فضای کاربری خود را مشاهده نمایید. در این برنامه شما با راحتی با استفاده از تکنیک Drop کردن می توانید فایل و دیتا مورد نظر خود را به سرور منتقل کنید.



اجرای کار:

شما با استفاده از دستور `module avail` می توانید لیست ماژول های نصب شده بر روی سرور را مشاهده کنید. همچنین `module list` لیست ماژول هایی که اخیرا استفاده شده است را نشان می دهد.

[username@master ~]\$ module avail

```
----- /opt/hpc2/not_easybuild/modules -----  
  
AutoDock/4  R/3.5.3/3.5.3  R/3.6.3/3.6.3  R/4.0.5/4.0.5  R/4.1.2/4.1.2  
  
----- /opt/hpc2/easybuild/modules/all ----- Autoconf/2.71-GCCcore-11.2.0  
Ninja/1.10.2-GCCcore-11.2.0      groff/1.22.4-GCCcore-11.2.0      (D)  
  
Automake/1.16.4-GCCcore-11.2.0      OpenBLAS/0.3.18-GCC-11.2.0      gzip/1.10-GCCcore-11.2.0  
  
Autotools/20210726-GCCcore-11.2.0      OpenMPI/4.1.1-GCC-11.2.0      help2man/1.47.16-GCCcore-10.  
  
BLIS/0.8.1-GCC-11.2.0      OpenSSL/1.1      (L)  help2man/1.48.3-GCCcore-11.2.0      (D)
```

[username@master ~]\$ module list

Currently Loaded Modules:

```
1) GCCcore/11.2.0      5) ncurses/6.2-GCCcore-11.2.0  9) XZ/5.2.5-GCCcore-11.2.0  13) Python/3.9.6-GCCcore-11.2.0  
2) zlib/1.2.11-GCCcore-11.2.0  6) libreadline/8.1-GCCcore-11.2.0  10) GMP/6.2.1-GCCcore-11.2.0  14) AutoDock/4  
3) binutils/2.37-GCCcore-11.2.0  7) Tcl/8.6.11-GCCcore-11.2.0  11) libffi/3.4.2-GCCcore-11.2.0  
4) bzip2/1.0.8-GCCcore-11.2.0  8) SQLite/3.36-GCCcore-11.2.0  12) OpenSSL/1.1
```

نمونه اجرای کار در R:

فرض کنید که بخواهیم فایل زیر را در R اجرا کنیم. محتوای فایل برای مثال مشابه زیر است و این فایل

به اسم R_example.R ذخیره شده است.

R_example.R

```
# Simple t-test between two equally large groups.  
# Let us generate some data.  
apples <- rnorm(20,mean=1.5,sd=.1)  
pears <- rnorm(20,mean=1.6,sd=.1)  
print(t.test(pears,apples))
```

برای اجرای برنامه ابتدا باید برای سیستم مشخصات کار تعریف کنیم. این فایل فرمت sh. دارد و سرور

مطابق با آن، دستورات را اجرا می‌نماید. فایلی به نام test_job.sh با محتوای زیر ایجاد کنید. برای ایجاد این فایل

می‌توانید در نرم افزار MobaXTerm از قسمت Tools>Moba TextEditor استفاده کنید و فایل را به صورت

sh. ذخیره نمایید.

test_job.sh

#!/bin/bash

Shebang

```
#SBATCH --job-name=test          # Job name (limit character usage)
#SBATCH --mail-type=ALL          # Options: ALL, NONE, BEGIN, END, FAIL, REQUEUE
#SBATCH --time=00:01:05          # time allocated to this job FORMAT: dd-hh:mm:ss
#SBATCH --output=r_example.out    # batch script's standard output file
#SBATCH -e Error.err              # batch script's standard error file
#SBATCH --nodes=1                # number of nodes allocated to this job
#SBATCH --mem=10MB                # memory allocated to each node
```

Requirement

```
pwd
module purge
Rscript R_example.R
```

Commands

در این فایل، خط اول به مفسر کار اشاره می کند:

#!/bin/bash

در صورت اجرای برنامه پایتون این خط نیز باید اضافه شود.

#!/usr/bin/env python

ملزمومات کار می تواند با استفاده از خطوطی که با دستور SBATCH آغاز می شود در داخل فایل اجرا

قرار بگیرد. این دستورات باید بلافاصله بعد از خط #!/bin/bash اضافه شود.

- 1) #SBATCH --job-name=test # Job name (limit character usage)
- 2) #SBATCH --mail-type=ALL # Options: ALL, NONE, BEGIN, END, FAIL, REQUEUE
- 3) #SBATCH --time=00:01:05 # time allocated to this job FORMAT: dd-hh:mm:ss
- 4) #SBATCH --output= r_example.out # batch script's standard output file
- 5) #SBATCH -e Error.err # batch script's standard error file
- 6) #SBATCH --nodes=1 # number of nodes allocated to this job
- 7) #SBATCH --mem=10MB # memory allocated to each node

در این فایل در خط اول نام کار تعیین می‌شود. خط دوم زمان ارسال اعلان های را به شما مشخص می‌کند. خط سوم مربوط به زمان و ساعت است. خط چهارم مشخص کننده نام فایل خروجی کار است. در خط پنجم نام فایلی که خطاهای احتمالی اجرا را گزارش می‌دهد مشخص شده است. خط ششم متغیر نود (nodes) تعداد گره‌های پردازشی را مشخص می‌کند. در صورت درخواست تعداد بیشتری گره ممکن است زمان بیشتری در صف انتظار بمانید. و در آخر در خط هفتم میزان حافظه مورد نیاز بر هر نود مشخص شده است.

برای کاربردهایی multithreadin شما می‌توانید درخواست هسته بیشتری بر روی یک نود بنمایید. که این مقادیر با توجه به ظرفیت سیستم تعیین می‌گردند.

- 1) #SBATCH --cpus-per-task= <N> # to request more cores on a single node
- 2) #SBATCH --ntasks-per-node=<X>

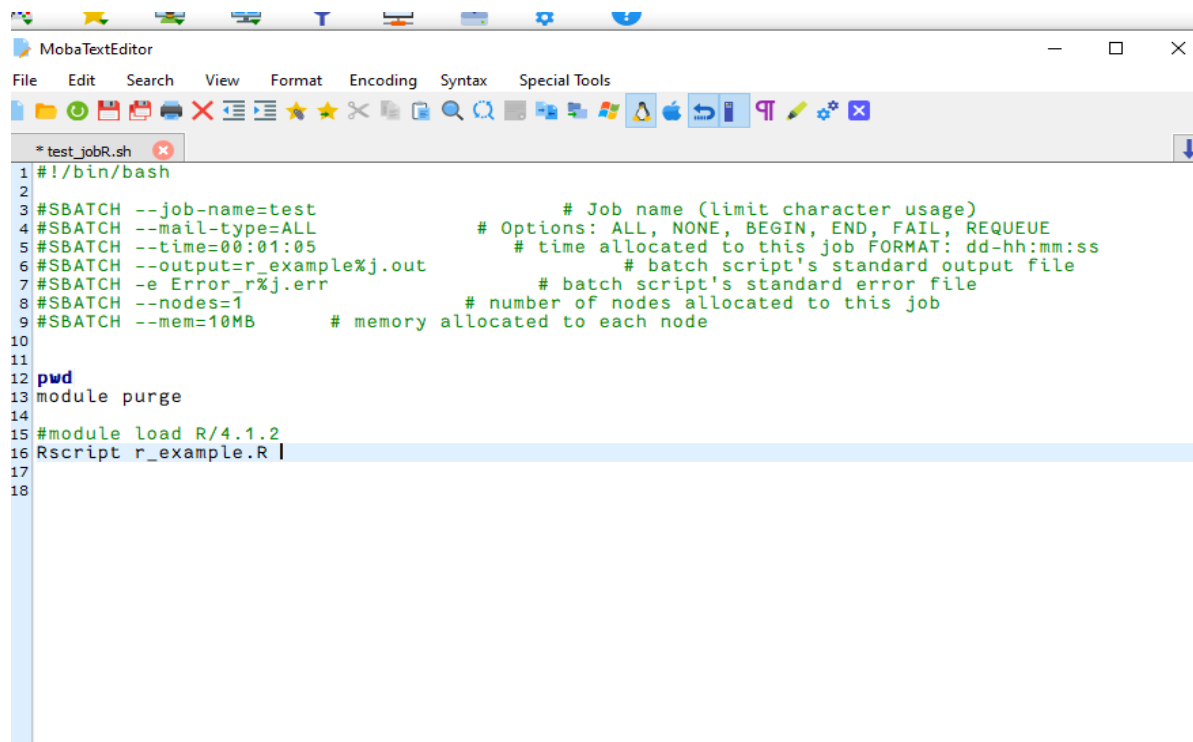
تا این قسمت از دستورات برای اجرای تمام برنامه‌ها مشترک است. در انتهای فایل کامند های مربوط به اجرای برنامه مورد نظر آورده می‌شود. که در اینجا برنامه R فراخوانی شده است.

pwd

module purge

Rscript R_example.R

در خط دوم تمام مازول ها unload شد و سیستم به حالا پایه بر میگردد. در خط سوم برنامه مورد نظر برای اجرا نوشته می‌شود. لازم به ذکر است برای اجرای برنامه R نیاز به فراخوانی آن وجود ندارد. نمای کلی برنامه به صورت شکل زیر است.



```
1#!/bin/bash
2
3#SBATCH --job-name=test                # Job name (limit character usage)
4#SBATCH --mail-type=ALL                # Options: ALL, NONE, BEGIN, END, FAIL, REQUEUE
5#SBATCH --time=00:01:05                # time allocated to this job FORMAT: dd-hh:mm:ss
6#SBATCH --output=r_example%j.out      # batch script's standard output file
7#SBATCH -e Error_r%j.err              # batch script's standard error file
8#SBATCH --nodes=1                     # number of nodes allocated to this job
9#SBATCH --mem=10MB                    # memory allocated to each node
10
11
12pwd
13module purge
14
15#module load R/4.1.2
16Rscript r_example.R
17
18
```

در انتها در صفحه command line برای اجرا دستور زیر نوشته می شود. توجه داشته باشید که هر دوی فایل های R و sh. در یک مسیر باشند.

```
[username@master ~]$ sbatch test_job.sh
```

با دستور squeue از روند اجرای کار خود آگاه می شوید.

```
[username@master ~]$ squeue
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
886	main	GE	user_name	R	1:37:58	1	node02

که در آن st حالت برنامه که در حال اجرا یا انتظار است را مشخص می کند. R نشان دهنده اجرا برنامه و PD نشان دهنده حالت انتظار است.

نمونه اجرای کار در پایتون

برای اجرای فایل پایتون در سامانه ابتدا لازم است محیط مجازی خود را ساخته و سپس کتابخانه های مورد نظر را در این محیط نصب نمایید. بدین منظور ابتدا ماژول پایتون در داخل سامانه با دستور زیر فراخوانی می شود.

```
[username@master ~]$ module load Python
```

سپس با استفاده از دستور

```
[username@master ~]$ virtualenv my_env_name
```

محیط جدید در داخل فضای کاربری شما ایجاد می شود. با استفاده از دستور زیر محیط برای شما فعال می شود.

```
[username@master ~]$ source my_env_name /bin/activate
```

توجه داشته باشید که برای نصب پکیج های جدید، قبل از فعالسازی محیط مجازی ساخته شده، در ابتدا باید ماژول پایتون فراخوانی شود.

```
[username@master ~]$ module load Python
```

```
[username@master ~]$ source my_env_name /bin/activate
```

سپس در این محیط می توانید با دستور pip کتابخانه های مختلف را نصب نمایید.

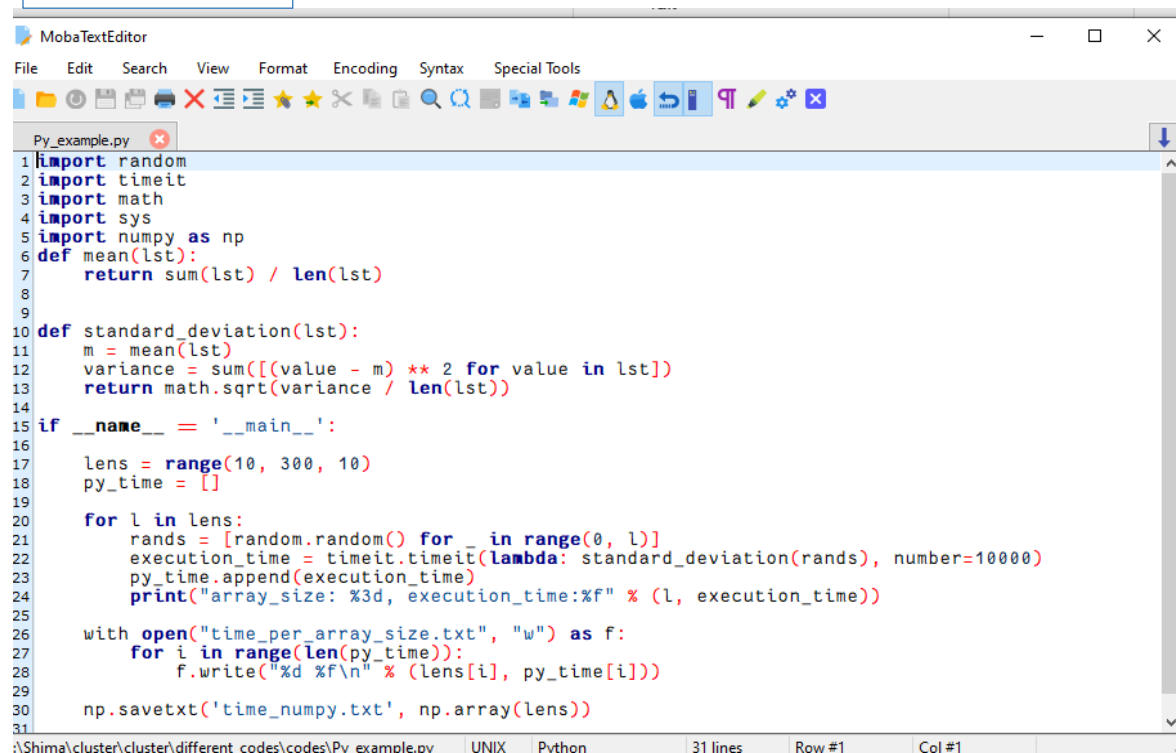
```
(my_env_name) [username@master ~]$ pip install pandas
```

برای خارج شدن از محیط از دستور غیر فعال کردن استفاده می شود.

```
(my_env_name) [username@master ~]$ deactivate
```

فرض کنید می خواهیم برنامه نوشته شده در پایتون به نام Py_example.py را در محیط پایتون اجرا کنید. محتوای فایل برای مثال مشابه زیر است.

Py_example.py



```
1 import random
2 import timeit
3 import math
4 import sys
5 import numpy as np
6 def mean(lst):
7     return sum(lst) / len(lst)
8
9
10 def standard_deviation(lst):
11     m = mean(lst)
12     variance = sum([(value - m) ** 2 for value in lst])
13     return math.sqrt(variance / len(lst))
14
15 if __name__ == '__main__':
16     lens = range(10, 300, 10)
17     py_time = []
18
19     for l in lens:
20         rands = [random.random() for _ in range(0, l)]
21         execution_time = timeit.timeit(lambda: standard_deviation(rands), number=10000)
22         py_time.append(execution_time)
23         print("array_size: %3d, execution_time:%f" % (l, execution_time))
24
25     with open("time_per_array_size.txt", "w") as f:
26         for i in range(len(py_time)):
27             f.write("%d %f\n" % (lens[i], py_time[i]))
28
29     np.savetxt('time_numpy.txt', np.array(lens))
30
31
```

برای اجرای یک فایل پایتون ، فایل اجرای sh. آن مطابق با توضیحات گفته شده در نمونه فایل R آماده می شود. توجه داشته باشید که در این فایل طبق مطلبی که در بالا گفته شد، بعد از خط اول باید خط اجرای پایتون (#!/usr/bin/env python) اضافه شود و هم چنین قسمت کامند آن که مربوط به فراخوانی ماژول هست متفاوت است.

Test_python.sh

```
#!/bin/bash

#!/usr/bin/env python

#SBATCH --job-name=test          # Job name (limit character usage)
#SBATCH --mail-type=ALL          # Options: ALL, NONE, BEGIN, END, FAIL, REQUEUE
#SBATCH --time=00:01:05          # time allocated to this job FORMAT: dd-hh:mm:ss
#SBATCH --output=Py_example.out  # batch script's standard output file
#SBATCH -e Error.err             # batch script's standard error file
#SBATCH --nodes=1               # number of nodes allocated to this job
#SBATCH --mem=10MB              # memory allocated to each node

pwd

module purge

module load Python

./ my_env_name /bin/python3.9 Py_example.py
```

برای اجرا در سامانه بعد از فراخوانی پایتون و فال کردن محیط مجازی فایل sh. را اجرا می کنیم.

```
[username@master ~]$ module load Python
```

```
[username@master ~]$ source my_env_name /bin/activate
```

```
(my_env_name) [username@master ~]$ sbatch test_python.sh
```

نمونه اجرای AutoDock

برای اجرای برنامه AutoDock همانطور که گفته شد در داخل فایل sh. بعد از اضافه کردن ملزومات و مشخصات کار، ابتدا باید برنامه AutoDock فراخوانی شود و سپس با استفاده از دستور srun کامندهای مربوط به آن اجرا می شود.

test_autodock.sh

```
#!/bin/bash
#
#SBATCH --job-name=test
#SBATCH --output=job-%j.log.txt
#SBATCH --nodes=1

module load AutoDock
srun ./autodock_example.sh
```